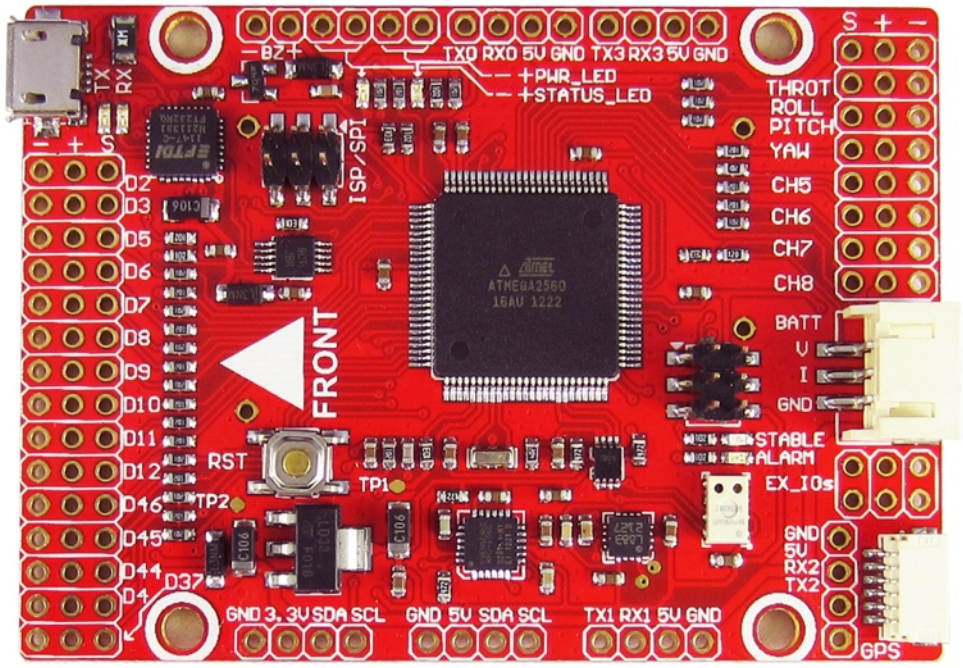
Getting Started with MultiWii Mega All-in-one Flight Controller



MultiWii and MegaPirate are open-source software for multi-rotors. MultiWii Mega is the hardware installed MultiWii or MegaPirate used to control Multi-rotors. Special thanks to MultWii and MegaPirate's contributors for their great work!

Main Features

CPU & software:

- ATmega2560 16MHz 5V
- MultiWii / MegaPirateNG Firmware Supported (PC/Mac)

Sensors:

- Acc & Gyro: MPU6050 (a real progress for having a perfect stabilization)
- Mag: HMC5883L (for "Heading hold", "Carefree" and some GPS functions)
- Baro: MS5611-01BA01 (for "ALT Hold", greater precision than BMP085)

Others:

- Up to 8 motors output
- Up to 8 input for RC transmitters from 4 to 8 channels
- 4 Serial ports(1 for PC/Mac connection, 1 for GPS module, 1 for telemetry)
- 2 servos output for 2-axis gimbal system
- 1 servo output for camera trig
- 4 Analog I/O for extended devices
- On board lipo monitor (support 3S, 4S lipo)
- On board FT232RQ USB to UART converter (connect to PC/Mac directly with the USB cable)
- 3.3V and 5V I2C (for sensors like sonar, optical flow sensor board)
- Buzzer (low battery indicator, control feedback)
- LED indicators (power LED, status LED, stable LED and alarm LED)
- Match the standard of RoHS

Dimensions:

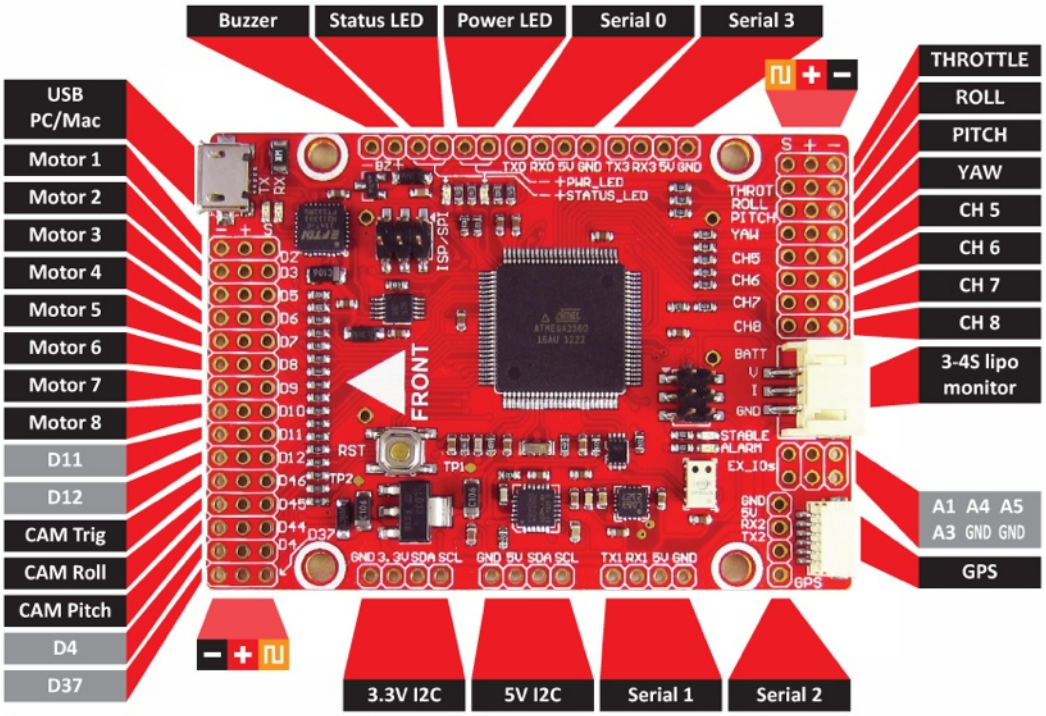
- 75 x50mm
- 45x45mm mounting holes（hole diameter: 3mm）

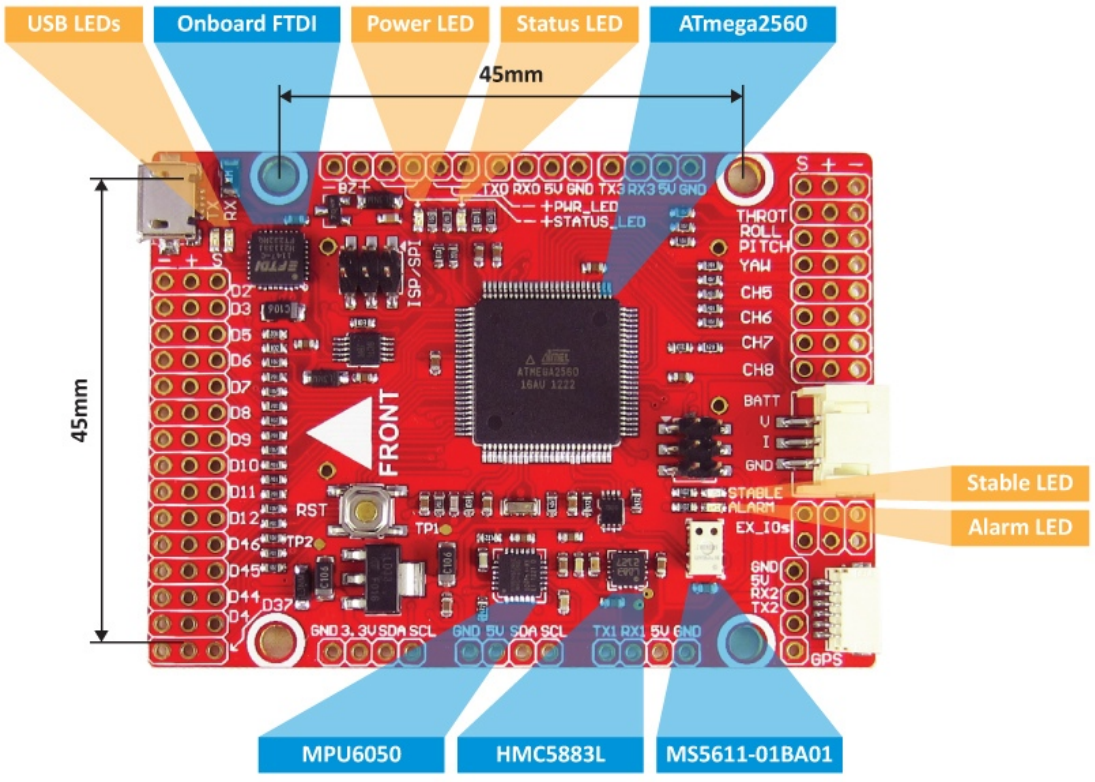
Special:

- 4 more digital I/O for extend motors or servos (up to 10 motors and 3-axis gimbal system)
- 4 more analog I/O for extend devices

For MultiWii Fans

1 Connect Diagram





InnovWorks

MultiWii Mega (ATmega2560)

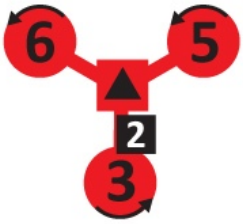
Motor in CCW

Motor in CW

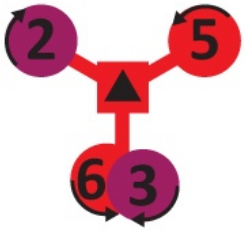
Servo

FRONT

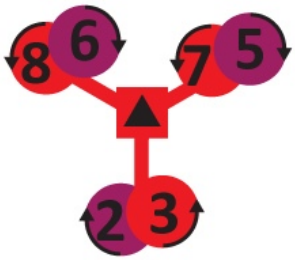
BICOPTER



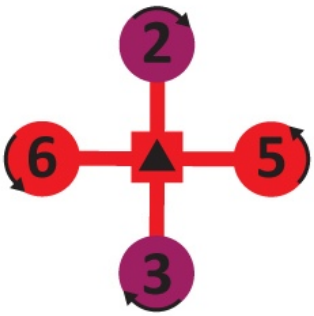
TRICOPTER



Y4



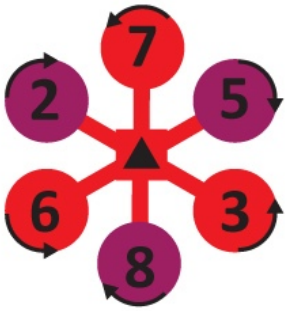
Y6



QUAD +

QUAD X

X8



HEXA +

HEXA X

OCTO +

OCTO X

2 Sensor Config

For users of MultiWii 2.1 or later : (uncomment MultiWiiMega on the list)

```
//#define CRIUS_SE        // Crius MultiWii SE
//#define OPENLRSv2MULTI  // OpenLRS v2 Multi Rc Receiver board including ITG3205 and ADXL345
//#define BOARD_PROTO_1   // with MPU6050 + HMC5883L + MS baro
//#define BOARD_PROTO_2   // with MPU6050 + slave  MAG3110 + MS baro
//#define GY_80           // Chinese 10 DOF with  L3G4200D ADXL345 HMC5883L BMP085, LLC
//#define GY_85           // Chinese 9 DOF with  ITG3205 ADXL345 HMC5883L LLC
//#define GY_86           // Chinese 10 DOF with  MPU6050 HMC5883L MS5611, LLC
//#define GY_521          // Chinese 6  DOF with  MPU6050, LLC
//#define INNOVWORKS_10DOF // with ITG3200, BMA180, HMC5883, BMP085 available here http://www.diymulticopter.com
//#define INNOVWORKS_6DOF // with ITG3200, BMA180 available here http://www.diymulticopter.com
#define MultiWiiMega      // MEGA + MPU6050+HMC5883L+MS5611 available here http://www.diymulticopter.com
//#define PROTO_DIY       // 10DOF mega board
//#define IOI_MINI_MULTIWII// www.bambucopter.com
//#define Bobs_6DOF_V1     // BobsQuads 6DOF V1 with ITG3200 & BMA180
//#define Bobs_9DOF_V1     // BobsQuads 9DOF V1 with ITG3200, BMA180 & HMC5883L
//#define Bobs_10DOF_BMP_V1 // BobsQuads 10DOF V1 with ITG3200, BMA180, HMC5883L & BMP180 - BMP180 is software compatible witl
//#define FLYDUINO_MPU
//#define CRIUS_AIO_PRO_V1
//#define DESQUARED6DOFV2GO  // DEsquared V2 with ITG3200 only
//#define DESQUARED6DOFV4    // DEsquared V4 with MPU6050
//#define LADYBIRD
//#define MEGAWAP_V2_STD     // available here: http://www.multircshop.com                  <- confirmed by Alex
//#define MEGAWAP_V2_ADV
//#define HK_MultiWii_SE_V2  // Hobbyking board with MPU6050 + HMC5883L + BMP085
//#define HK_MultiWii_328P   // Also labeled "Hobbybro" on the back.  ITG3205 + BMA180 + BMP085 + NMC5583L + DSM2 Connector (
```

```
/**************************    independent sensors    *********************************/
  /* leave it commented if you already checked a specific board above */
  /* I2C gyroscope */
```

For users of MultiWii 2.0 : (The Multwii Mega have the same sensor configuration as FREEIMU_v043)

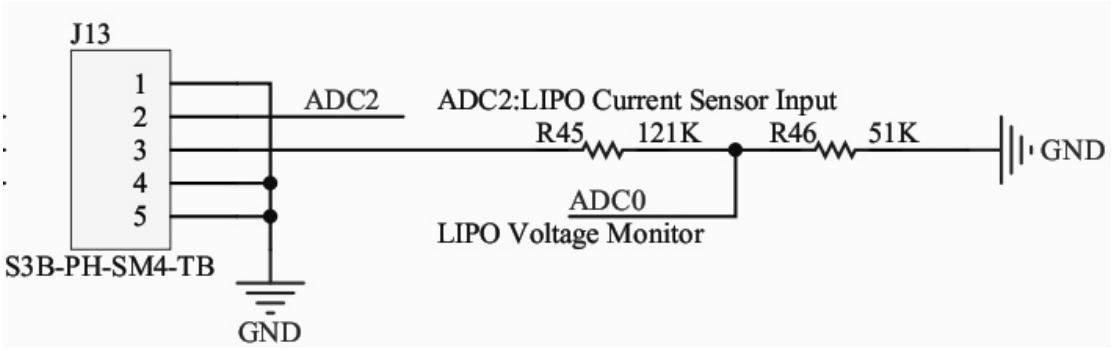
```
/*****************************************************************************************/
/*****************         boards and sensor definitions          ******************/
/*****************************************************************************************/

 /**************************    Combined IMU Boards    *********************************/
   /* if you use a specific sensor board:
      please submit any correction to this list.
        Note from Alex: I only own some boards, for other boards, I'm not sure, the info was gathered via rc forums, be
   //#define FFIMUv1          // first 9DOF+baro board from Jussi, with HMC5843                  <- confirmed by Alex
   //#define FFIMUv2          // second version of 9DOF+baro board from Jussi, with HMC5883      <- confirmed by Alex
   //#define FREEIMUv1        // v0.1 & v0.2 & v0.3 version of 9DOF board from Fabio
   //#define FREEIMUv03       // FreeIMU v0.3 and v0.3.1
   //#define FREEIMUv035      // FreeIMU v0.3.5 no baro
   //#define FREEIMUv035_MS   // FreeIMU v0.3.5_MS                                              <- confirmed by Alex
   //#define FREEIMUv035_BMP  // FreeIMU v0.3.5_BMP
   //#define FREEIMUv04       // FreeIMU v0.4 with MPU6050, HMC5883L, MS561101BA                <- confirmed by Alex
   #define FREEIMUv043      // same as FREEIMUv04 with final MPU6050 (with the right ACC scale)
   //#define NANOWII          // the smallest multiwii FC based on MPU6050 + pro micro based proc <- confirmed by Alex
   //#define PIPO             // 9DOF board from erazz
   //#define QUADRINO         // full FC board 9DOF+baro board from witespy  with BMP085 baro    <- confirmed by Alex
```

Or you can define the sensors by adding this below in the “config.h” file :

#define MPU6050
#define HMC5883
#define MS561101BA
#define ACC_ORIENTATION(X, Y, Z) {accADC[ROLL] = -X; accADC[PITCH] = -Y; accADC[YAW] = Z;}
#define GYRO_ORIENTATION(X, Y, Z) {gyroADC[ROLL] = Y; gyroADC[PITCH] = -X; gyroADC[YAW] = -Z;}
#define MAG_ORIENTATION(X, Y, Z) {magADC[ROLL] = X; magADC[PITCH] = Y; magADC[YAW] = -Z;}
#define MPU6050_I2C_AUX_MASTER
#undef INTERNAL_I2C_PULLUPS

3 Battery Monitor

Two resistors used to monitoring the battery voltage, the schematic as below:



If you use a 3S lipo battery, define as below in the file “config.h”:

```
/************************************************************************/
/****          battery voltage monitoring                          ****/
/************************************************************************/
  /* for V BAT monitoring
     after the resistor divisor we should get [0V;5V]->[0;1023] on analog V_BATPIN
     with R1=33k and R2=51k
     vbat = [0;1023]*16/VBATSCALE
     must be associated with #define BUZZER ! */
  #define VBAT             // uncomment this line to activate the vbat code
  #define VBATSCALE     97 // change this value if readed Battery voltage is different than real voltage
  #define VBATLEVEL1_3S 111 // 14.8V
  #define VBATLEVEL2_3S 107 // 14.3V
 // #define VBATLEVEL3_3S 99  // 9.9V
  #define NO_VBAT       16 // Avoid beeping without any battery
```

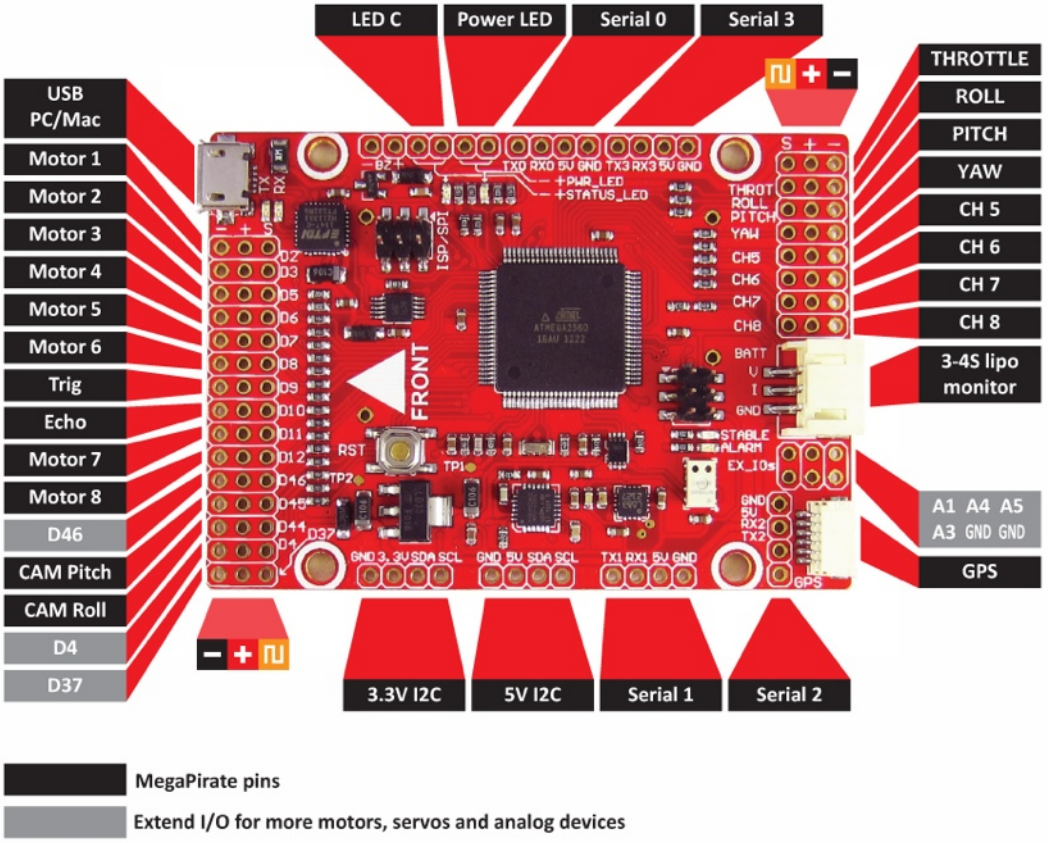
If you use a 4S lipo battery, define as below in the file “config.h”:

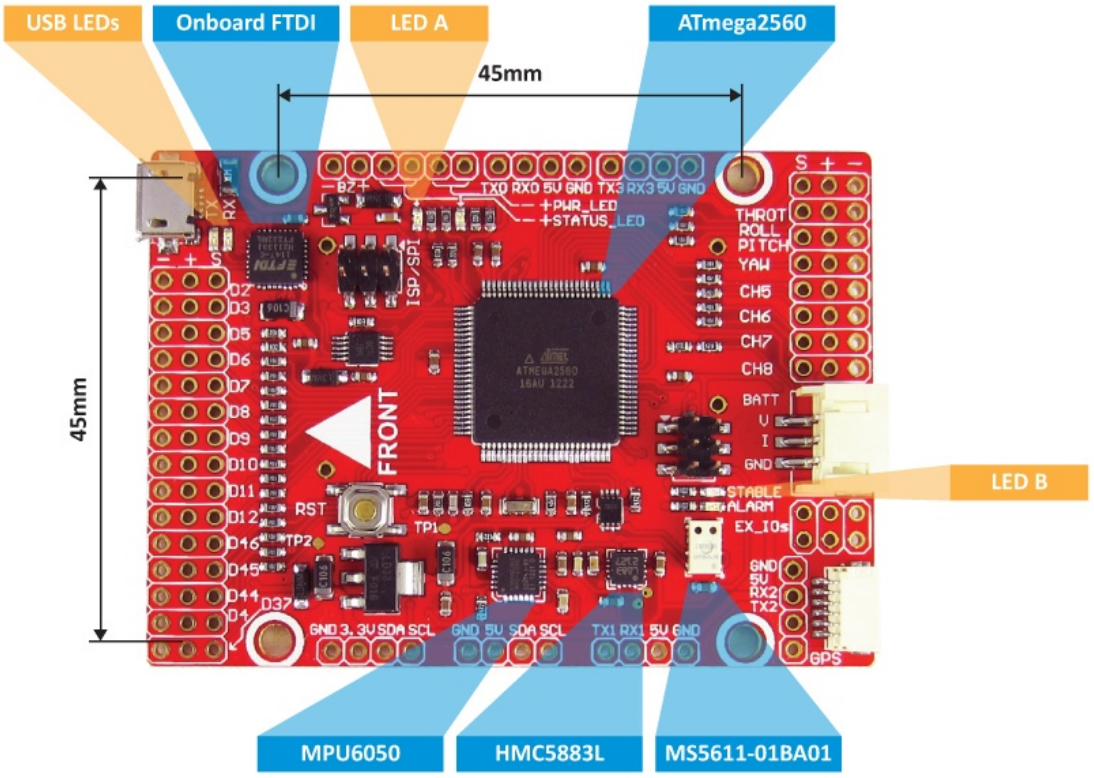
```
/************************************************************************/
/****          battery voltage monitoring                          ****/
/************************************************************************/
  /* for V BAT monitoring
     after the resistor divisor we should get [0V;5V]->[0;1023] on analog V_BATPIN
     with R1=33k and R2=51k
     vbat = [0;1023]*16/VBATSCALE
     must be associated with #define BUZZER ! */
  #define VBAT             // uncomment this line to activate the vbat code
  #define VBATSCALE     97 // change this value if readed Battery voltage is different than real voltage
  #define VBATLEVEL1_3S 148 // 14.8V
  #define VBATLEVEL2_3S 143 // 14.3V
 // #define VBATLEVEL3_3S 99  // 9.9V
  #define NO_VBAT       16 // Avoid beeping without any battery
```

A current sensor should be used to monitor the battery current, please refer to the current sensor’s user manual for more details.

For MegaPirate Fans

1 Connect Diagram





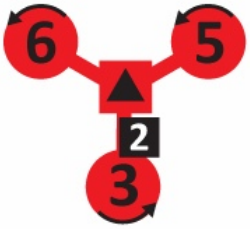
MegaPirate

Motor in CCW

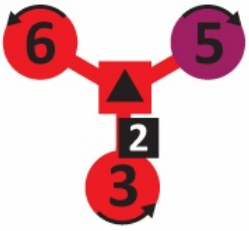
Motor in CW

Servo

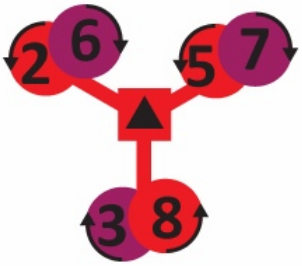
FRONT



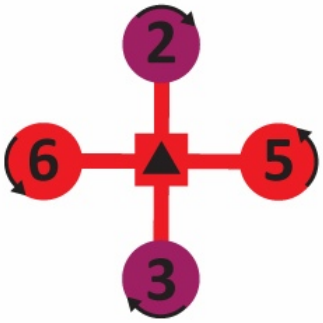
TRICOPTER



TRICOPTER



Y6

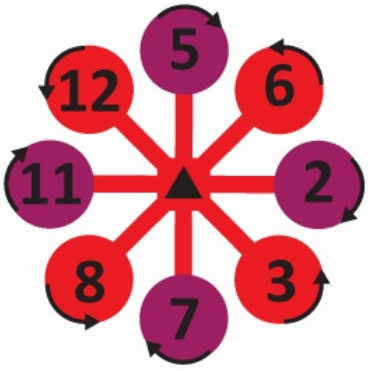


QUAD +

QUAD X

HEXA +

HEXA X



OCTO +

OCTO X

2 Tweak the code

2.1 Select Sensor Board and Sensor Config

APM_Config.h

```
// Select your sensor board
#define PIRATES_SENSOR_BOARD PIRATES_DROTEK_10DOF_MPU
/*
        PIRATES_ALLINONE
        PIRATES_FFIMU
        PIRATES_FREEIMU
        PIRATES_BLACKVORTEX
        PIRATES_FREEIMU_4                                       // New FreeIMU 0.4.1 with MPU6000, MS5611 and 5883L
        PIRATES_DROTEK_10DOF_MPU                // MPU6000, MS5611 and 5883L
        PIRATES_CRIUS_AIO_PRO_V1    // Crius AllInOne Pro v1
*/
```

config.h

```
#if CONFIG_APM_HARDWARE == APM_HARDWARE_PIRATES
        #ifndef LED_SEQUENCER
                #define LED_SEQUENCER DISABLED
        #endif

        #ifndef PIRATES_SENSOR_BOARD
                #define PIRATES_SENSOR_BOARD PIRATES_ALLINONE
        #endif

        #if PIRATES_SENSOR_BOARD == PIRATES_FREEIMU_4
                #define CONFIG_IMU_TYPE   CONFIG_IMU_MPU6000_I2C
                #define CONFIG_BARO     AP_BARO_MS5611_I2C
                #define MAG_ORIENTATION ROTATION_YAW_270
                #define GYRO_ADDR 0x68
        #elif PIRATES_SENSOR_BOARD == PIRATES_DROTEK_10DOF_MPU
                #define CONFIG_IMU_TYPE   CONFIG_IMU_MPU6000_I2C
                #define CONFIG_BARO     AP_BARO_MS5611_I2C
                #define MAG_ORIENTATION ROTATION_YAW_180
                #define GYRO_ADDR 0x68
```

2.2 GPS configuration

For our MTK3329 GPS Module (Firmware: 115200bps, 10Hz), tweak the code as below:
APM_Config.h

```
// For BlackVortex, just set PIRATES_SENSOR_BOARD as PIRATES_BLACKVORTEX, GPS will be selected automatically
#define GPS_PROTOCOL GPS_PROTOCOL_NMEA
/*
        GPS_PROTOCOL_NONE       without GPS
        GPS_PROTOCOL_NMEA
        GPS_PROTOCOL_SIRF
        GPS_PROTOCOL_UBLOX      <<< Select this for UBLOX LEA-6 (CRIUS GPS boards and others)
        GPS_PROTOCOL_MTK16
        GPS_PROTOCOL_BLACKVORTEX
        GPS_PROTOCOL_AUTO       auto select GPS
*/

#define SERIAL0_BAUD                    115200 // Console port
#define SERIAL2_BAUD                    115200 // GPS port
#define SERIAL3_BAUD                    57600  // Telemetry (MAVLINK) port
```

Other GPS Module setting, please contact your supplier for more information.

2.3 Frame Config

The default frame configuration as below, if you have different frame type, changes the macro definition!

```
#define FRAME_CONFIG QUAD_FRAME
/*
        QUAD_FRAME
        TRI_FRAME
        HEXA_FRAME
        Y6_FRAME
        OCTA_FRAME
        OCTA_QUAD_FRAME
        HELI_FRAME
*/

#define FRAME_ORIENTATION X_FRAME
/*
        PLUS_FRAME
        X_FRAME
        V_FRAME
*/
```